Multi-Space Neural Radiance Fields

Ze-Xin Yin  Jiaxiong Qiu  Ming-Ming Cheng  Bo Ren*
VCIP, CS, Nankai University
{Zexin.Yin.cn, qiujiaxiong727}@gmail.com, {cmm, rb}@nankai.edu.cn

Abstract

Existing Neural Radiance Fields (NeRF) methods suffer from the existence of reflective objects, often resulting in blurry or distorted rendering. Instead of calculating a single radiance field, we propose a multi-space neural radiance field (MS-NeRF) that represents the scene using a group of feature fields in parallel sub-spaces, which leads to a better understanding of the neural network toward the existence of reflective and refractive objects. Our multi-space scheme works as an enhancement to existing NeRF methods, with only small computational overheads needed for training and inferring the extra-space outputs. We demonstrate the superiority and compatibility of our approach using three representative NeRF-based models, i.e., NeRF, Mip-NeRF, and Mip-NeRF 360. Comparisons are performed on a newly constructed dataset consisting of 25 synthetic scenes and 7 real captured scenes with complex reflection and refraction, all having 360-degree viewpoints. Extensive experiments show that our approach significantly outperforms the existing single-space NeRF methods for rendering high-quality scenes concerned with complex light paths through mirror-like objects. Our code and dataset will be publicly available at https://zx-yin.github.io/msnerf.

1. Introduction

Neural Radiance Fields (NeRF) [25] and its variants are refreshing the community of neural rendering, and the potential for more promising applications is still under exploration. NeRF represents scenes as continuous radiance fields stored by simple Multi-layer Perceptrons (MLPs) and renders novel views by integrating the densities and radiance, which are queried from the MLPs by points sampled along the ray from the camera to the image plane. Since its first presentation [25], many efforts have been investigated to enhance the method, such as extending to unbounded scenes [2, 50], handling moving objects [29, 30, 37], or reconstructing from pictures in the wild [6, 21, 35, 49].

*Bo Ren is the corresponding author.

(a) Mip-NeRF 360 [2], SSIM=0.825  (b) Our Model, SSIM=0.881

Figure 1. (a) Though Mip-NeRF 360 can handle unbounded scenes, it still suffers from reflective surfaces, as the virtual images violate the multi-view consistency, which is of vital importance to NeRF-based methods. (b) Our method can help conventional NeRF-like methods learn the virtual images with little extra cost.

However, rendering scenes with mirrors is still a challenging task for state-of-the-art NeRF-like methods. One of the principle assumptions for the NeRF method is the multi-view consistency property of the target scenes [16, 20, 36]. When there are mirrors in the space, if one allows the viewpoints to move 360-degree around the scene, there is no consistency between the front and back views of a mirror, since the mirror surface and its reflected virtual image are only visible from a small range of views. As a result, it is often required to manually label the reflective surfaces in order to avoid falling into sub-optimal convergences [12].

In this paper, we propose a novel multi-space NeRF-based method to allow the automatic handling of mirror-like objects in the 360-degree high-fidelity rendering of scenes without any manual labeling. Instead of regarding the Euclidean scene space as one single space, we treat it as composed of multiple virtual sub-spaces, whose composition changes according to location and view direction. We show that our approach using such a multi-space decomposition leads to successful handlements of complex reflections and refractions where the multi-view consistency is heavily violated in the Euclidean real space. Furthermore, we show that the above benefits can be achieved by designing a low-cost multi-space module and replacing the original output layer with it. Therefore, our multi-space approach serves as
a general enhancement to the NeRF-based backbone, equipping most NeRF-like methods with the ability to model complex reflection and refraction, as shown in Fig. 1.

Existing datasets have not paid enough attention to the 360-degree rendering of scenes containing mirror-like objects, such as RFFR [12] just has forward-facing scenes, and the Shiny dataset in [42] with small viewpoints changes and cannot exhibit viewpoint-dependent effects in large angle scale. Therefore we construct a novel dataset dedicated to evaluation for the 360-degree high-fidelity rendering of scenes containing complex reflections and refractions. In this dataset, we collect 25 synthesized scenes and 7 captured real-world scenes. Each synthesized scene consists of 120 images of 360-degree around reflective or refractive objects, with 100 randomly split for training, 10 for validation, and 10 for evaluation. Each real-world scene is captured randomly around scenes with reflective and refractive objects, consisting of 62 to 118 images, and organized under the convention of LLFF [24]. We then demonstrate the superiority and compatibility of our approach by comparisons, using three representative baseline models, i.e., NeRF [25], Mip-NeRF [1], and Mip-NeRF 360 [2], with and without our multi-space module. Experiments show that our approach improves performance by a large margin both quantitatively and qualitatively on scenes with reflection and refraction. Our main contributions are as follows:

- We propose a multi-space NeRF method that automatically handles mirror-like objects in 360-degree high-fidelity scene rendering, achieving significant improvements over existing representative baselines both quantitatively and qualitatively.

- We design a lightweight module that can equip most NeRF-like methods with the ability to model reflection and refraction with small computational overheads.

- We construct a dataset dedicated to evaluation for the 360-degree high-fidelity rendering of scenes containing complex reflections and refractions, including 25 synthesized scenes and 7 real captured scenes.

2. Related works

Coordinate-based novel view synthesis. NeRF [25] has bridged the gap between computer vision and computer graphics, and reveals a promising way to render high-quality photorealistic scenes with only posed images. The insights and the generalization ability of this scheme also facilitate various tasks both in CV and CG, i.e., 3D reconstruction [28, 40], 3D-aware generation [4, 15, 27], 3D-aware edition [39, 47], and avatar reconstruction and manipulation [5, 18, 52]. Besides, researchers have made great efforts to enhance this scheme. Mip-NeRF [1] enhances the anti-aliasing ability of NeRF by featuring 3D conical frustum using integrated positional encoding. [14, 23] adapt this scheme to HDR images. [2, 50] extend NeRF and its variants to unbounded scenes. There are also many works trying to speed up the training and inference speed using explicit or hybrid representations [5, 7, 10, 26, 32, 34, 46].

Glossy materials with high specular have a great influence on NeRF-like methods, [38] is inspired by precomputation-based techniques [31] in computer graphics to represent and render viewpoint-dependent specular and reflection, but it fails to handle mirror-like reflective surfaces because the virtual images cannot be treated as textures. Guo et al. [12] propose to decompose reflective surfaces into a transmitted part and reflected part, which is the most relevant work to ours, but such decomposition cannot handle 360-degree views with mirror-like objects because the virtual images have no difference from real objects until the viewpoint moves beyond a certain angle.


Commonly used datasets. Researchers have introduced or constructed many different datasets to facilitate the development of NeRF-based methods in various tasks. Mildenhall et al. [25] collect a dataset containing eight rendered sets of posed images about eight objects separately, and eight real captured forward-facing scenes with the camera poses and intrinsics estimated by COLMAP [33]. Nevertheless, these scenes lack reflection and refraction, which are very common. Wizadwongsa et al. [42] propose a dataset, namely Shiny, that contains eight more challenging scenes to test NeRF-like methods on viewpoint-dependent effects, but they are captured in a roughly forward-facing manner. Verbin et al. [38] create a dataset of six glossy objects, namely Shiny Blender, which are rendered under similar conditions as done in NeRF to test methods in modeling more complex materials. For unbounded scenes, Barron et al. [2] construct a dataset consisting of 5 outdoor scenes and 4 indoor scenes, while Zhang et al. [50] adopt Tanks and Temples (T&T) dataset [19] and the Light Field dataset [48]. Bemana et al. [3] capture a dataset consisting of refractive objects, which is composed of four scenes with cameras moving in a large range. Guo et al. [12] collect six forward-facing scenes with reflective and semi-transparent materials, which is, to date, the most relevant dataset to ours, but ours is much more challenging. DTU dataset [17] and Blended-MVS dataset [45] are commonly used as benchmarks for the evaluation of 3D reconstruction.
3. Method

3.1. Preliminaries: Neural Radiance Fields

Neural Radiance Fields (NeRF) [25] encodes a scene in the form of continuous volumetric fields into the weights of a multilayer perceptron (MLP), and adopts the absorption-only model in the traditional volumetric rendering to synthesize novel views. The training process only requires a sparse set of posed images and casts rays \( r(t) = o + td \) through the scene, where \( o \in \mathbb{R}^3 \) is the camera center and \( d \in \mathbb{R}^3 \) is the view direction, and the rays can be calculated by intrinsics and poses from the training data. Given these rays, NeRF samples a set of 3D points \( \{ p_i = o + t_i d \} \) by the distance to the camera \( t_i \) in the Euclidean space and projects these points to a higher dimensional space using the following function:

\[
\gamma(p) = [\sin(p), \cos(p), \ldots, \sin(2^{L-1}p), \cos(2^{L-1}p)]
\tag{1}
\]

where \( L \) is a hyperparameter and \( p \) is a sampled point.

Given the projected features \( \{ \gamma(p_i) \} \) and the ray direction \( d \), the MLP outputs the densities \( \{ \sigma_i \} \) and colors \( \{ c_i \} \), which are used to estimate the color \( C(r) \) of the ray using the quadrature rule reviewed by Max [22]:

\[
\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) c_i \tag{2}
\]

with \( T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \) and \( \delta_i = t_i - t_{i-1} \). Since the equation is differentiable, the model parameters can be optimized directly by Mean Squared Error (MSE) loss:

\[
L = \frac{1}{|R|} \sum_{r \in R} ||\hat{C}(r) - C(r)||_2^2 \tag{3}
\]

where \( R \) is a training batch of rays. Besides, NeRF also adopts a hierarchical sampling strategy to sample more points where higher weights are accumulated. With these designs, NeRF achieves state-of-the-art photorealistic results of novel view synthesis in most cases.

3.2. Multi-space Neural Radiance Field

The volumetric rendering equation and the continuous representation ability of MLPs do guarantee the success of NeRF-based methods in novel view synthesis, but as pointed out by previous works [12, 16, 20], there is also an unignorable property hidden in the training process that helps the convergence, which is the multi-view consistency. However, the multi-view consistency can be easily violated by any reflective surfaces. An example is shown in Fig. 2, when looking in front of a mirror one can observe the reflective virtual image as if there were an object behind it, but when looking from a side or backward, there is actually nothing behind the mirror. In practice, this means there will be completely conflictive training batches violating the fitting process of MLP.

To experimentally demonstrate the importance of multi-view consistency and its influence on the conventional NeRF network structure, We create two 360-degree toy scenes using an open source software Blender [8], each of which consists of 100 training images and 10 test images, training view examples are shown in Fig. 3a and Fig. 3b. The only difference between the two scenes is that we place a mirror-posed real object behind the mirror in the latter scene, but not in the former one. We train the vanilla NeRF separately on these toy scenes under the same setting and render some views from the test set as in Fig. 3c and Fig. 3d, which clearly shows that the vanished virtual image (i.e., violation to the multi-view consistency) in some views leads
the model to suboptimal results in reflection-related regions and produces blur in rendering. Interestingly, the conventional NeRF is still trying to fulfill the multi-view consistency assumption in the process. From the depth map in Fig. 3c, we can easily conclude that the conventional NeRF treats the viewed virtual image as a “texture” on the reflective surface, achieving a compromise between its principle assumption and the conflicts in training data, although the compromise leads to false understandings and worse rendering results of the real scenes.

Contrary to the conventional NeRF, inspired by the common perspective in Physics and Computer Graphics that reflective light can be viewed as “directly emitted” from its mirror-symmetric direction, from a possible “virtual source inside the virtual space in the mirror,” we build our novel multi-space NeRF approach on the following assumption:

**Assumption 1** At the existence of reflection and refraction, the real Euclidean scene space can be decomposed into multiple virtual sub-spaces. Each sub-space satisfies the multi-view consistency property.

It follows that the composition weights of the sub-spaces can change according to the spatial location and the view direction. Thus all sub-space contributes dynamically to the final render result. In this way, the violation of the multi-view consistency in real Euclidean space when there is a reflective surface can be overcome by placing the virtual images in certain sub-spaces only visible from certain views, as shown in Fig. 5.

### 3.3. Multi-Space module

A naive implementation of the multi-space NeRF network would be constructing the network using multiple tiny parallel MLPs, with each one representing one sub-space information, which, however, will largely increase parameters and has been proven to be tough to converge [32].

Besides, our experiments in Sec. 3.2 demonstrate that the current network structure of NeRF possesses the potential to understand our 3D scenes. Therefore, we propose a compact Multi-Space module (MS module), which takes advantage of the neural feature field scheme originally designed for memory saving [27], to sufficiently extract multi-space information from standard NeRF backbone network structures with only small computational overheads. Specifically, the MS module will replace the original output layer of the NeRF backbone. Below we describe the detailed architecture of our module.

As shown in Fig. 4, our MS module only modifies the output part of vanilla NeRF. Vanilla NeRF computes single density $\sigma$ and radiance $c$, for each position along a ray casting through the scene and performs volumetric rendering using Eq. (2) to get the accumulated color. On the contrary, our multi-head layer replaces the neural radiance field with the neural feature fields [27]. Specifically, the modified output layer gives $K$ densities $\{\sigma^k\}$ and features $\{f^k\}$ of $d$ dimension for each position along a ray with each pair corresponding to a sub-space, where $K$ and $d$ are hyper-
parameters for the total sub-space number and the feature dimension of the neural feature field, respectively.

We then integrate features along the ray in each sub-space to collect $K$ feature maps that encode the color information and visibility of each sub-space from a certain viewpoint. As all pixels are calculated the same way, we denote each pixel as $\{F^k\}$ for simplicity and describe the operation at the pixel level. Each pixel $\{F^k\}$ of the feature maps is calculated using:

$$\hat{F}^k(r) = \sum_{i=1}^{N} T^k_i (1 - \exp(-\sigma^k_i \delta)) t^k_i, \quad (4)$$

where the superscript $k$ indicates the sub-space that the ray casts through. The $k$-th density $\sigma^k_i$ and feature $t^k_i$ correspond to the $k$-th sub-space. $T^k_i = \exp(-\sum_{j=1}^{i-1} \sigma^k_j \delta_j)$ and $\delta_i = t_i - t_{i-1}$ are similarly computed as in Eq. (2).

Then, $\{F^k\}$ is decoded by two small MLPs, each with just one hidden layer. The first is a Decoder MLP that takes $\{F^k\}$ as input and outputs RGB vectors. The second is a Gate MLP that takes $\{F^k\}$ as input and outputs weights that control the visibility of certain sub-space. Specifically, we use:

$$\{F^k\} \xrightarrow{\Theta_D} \{C^k\}, \{F^k\} \xrightarrow{\Theta_G} \{w^k\}, \quad (5)$$

where $\Theta_D$ represents the Decoder MLP and $\Theta_G$ represents the Gate MLP. In the end, the MS module applies the softmax function to $\{w^k\}$ as the color contribution of each sub-space to form the final render results:

$$\hat{C}(r) = \frac{1}{\sum_{i=1}^{K} \exp(w^i)} \sum_{k=1}^{K} \exp(w^k) C^k. \quad (6)$$

Eq. (6) needs no additional loss terms compared with the vanilla NeRF method. As a result, the above light-weighted MS module is able to serve as an enhancement module onto the conventional NeRF-like backbone networks, and we will show that our approach achieves significant enhancements in Sec. 5.2.

4. Dataset

4.1. Existing datasets

We briefly revisit the commonly used or most relevant datasets to our task and list their properties in Tab. 1. As can be seen, there lacks a 360-degree benchmark for scenes with complex light paths, e.g., a glass of water in front of a mirror.

4.2. Our proposed dataset

As summarized in Sec. 4.1, there lacks a 360-degree dataset consisting of complex reflection and refraction to facilitate the related research. Therefore we collect a 360-degree dataset comprising 25 synthetic scenes and 7 real captured scenes.

For our synthesized part shown in Fig. 6a, we use an open source software Blender [8] and design our scenes with 3D models from BlenderKit, a community for sharing 3D models. As our dataset consists of complete scenes instead of single objects, we fix the height of our camera position with the camera looking at the center of the scene and moving the camera around a circle to render the whole scene.

For all our scenes, we uniformly sample 120 points along the circle and randomly choose 100 images for the training set, 10 for the validation set, and 10 for the test set. The constructed dataset features a wide variety of scenes containing reflective and refractive objects. We include a variety of complexity of light paths, controlled by the number and the layout of the mirror(s) in the scene, where the number of mirrors ranges from 1 up to tens of small pieces. Note that even a scene in our dataset with only one mirror is more challenging than RFFR [25], as our camera moves from the

<table>
<thead>
<tr>
<th>dataset</th>
<th>origin</th>
<th>applications</th>
<th>Type</th>
<th>viewpoints</th>
<th>properties</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Realistic Synthetic 360°</td>
<td>[25]</td>
<td>view synthesis</td>
<td>S</td>
<td>360-degree</td>
<td>non-Lambertian</td>
<td>8</td>
</tr>
<tr>
<td>Real Forward-Facing</td>
<td>[24, 25]</td>
<td>view synthesis</td>
<td>R</td>
<td>forward-facing</td>
<td>non-Lambertian</td>
<td>8</td>
</tr>
<tr>
<td>Shiny</td>
<td>[42]</td>
<td>view synthesis</td>
<td>R</td>
<td>forward-facing</td>
<td>high-specular, refraction</td>
<td>8</td>
</tr>
<tr>
<td>Tanks and Temples(T&amp;T)</td>
<td>[19]</td>
<td>view synthesis</td>
<td>R</td>
<td>360-degree</td>
<td>unbounded scenes</td>
<td>4</td>
</tr>
<tr>
<td>Mip-NeRF 360</td>
<td>[2]</td>
<td>view synthesis</td>
<td>R</td>
<td>360-degree</td>
<td>unbounded scenes</td>
<td>9</td>
</tr>
<tr>
<td>EikonalFields</td>
<td>[3]</td>
<td>view synthesis</td>
<td>R</td>
<td>large angle view</td>
<td>refraction</td>
<td>4</td>
</tr>
<tr>
<td>RFFR</td>
<td>[12]</td>
<td>view synthesis</td>
<td>R</td>
<td>forward-facing</td>
<td>reflection, semi-transparent</td>
<td>6</td>
</tr>
<tr>
<td>DTU</td>
<td>[17]</td>
<td>reconstruction</td>
<td>R</td>
<td>360-degree</td>
<td>non-Lambertian objects</td>
<td>15*</td>
</tr>
<tr>
<td>BlendedMVS</td>
<td>[45]</td>
<td>reconstruction</td>
<td>S</td>
<td>360-degree</td>
<td>non-Lambertian scenes</td>
<td>7*</td>
</tr>
<tr>
<td>Shiny Blender</td>
<td>[38]</td>
<td>view synthesis</td>
<td>S</td>
<td>360-degree</td>
<td>glossy materials</td>
<td>6</td>
</tr>
<tr>
<td>Ref-NeRF Real captured scenes</td>
<td>[13, 38]</td>
<td>view synthesis</td>
<td>R</td>
<td>360-degree</td>
<td>glossy materials</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1. Properties of a commonly used dataset for NeRF-based methods. ‘S’ and ‘R’ represent synthesized and real captured, respectively. We denote those unnamed datasets with the name of the methods. ‘*’ refers to the number of scenes commonly used by NeRF-based methods, as the original dataset contains more scenes than noted, and we do not take them into consideration.
and MS-Mip-NeRF

We build MS-Mip-NeRF [1] based experiments, we construct MS-Mip-NeRF based experiments, we carry out all the experiments following most default setting from [1, 2, 12, 25, 38], except that we use 1024 rays per batch and train 200k iterations for all experiments on all scenes.

We conduct three sets of experiments based on different datasets with different baselines and our modules of different scales. As our module is quite simple, we can scale our module by three hyperparameters, which we refer to as \( K \) for the sub-space number, \( d \) for the dimension of output features, and \( h \) for the hidden layer dimension of Decoder MLP and Gate MLP, respectively. To compare fairly, we carry out all the experiments following most default setting from [1, 2, 12, 25, 38], except that we use 1024 rays per batch and train 200k iterations for all experiments on all scenes. Experiment details are as follows.

We select three representative NeRF-based models as our baselines and integrate our modules with them. For NeRF [25] and Mip-NeRF [1] based experiments, we build MS-NeRF\(_S\) and MS-Mip-NeRF\(_S\) with hyperparameters \( \{ K = 6, d = 24, h = 48 \} \), and MS-NeRF\(_M\) and MS-Mip-NeRF\(_M\) with hyperparameters \( \{ K = 6, d = 48, h = 48 \} \), and MS-NeRF\(_B\) and MS-Mip-NeRF\(_B\) with hyperparameters \( \{ K = 8, d = 64, h = 64 \} \). For Mip-NeRF 360 [2] based experiments, we construct MS-Mip-NeRF 360 with hyperparameters \( \{ K = 8, d = 32, h = 64 \} \). Moreover, we provide a comparison with Ref-NeRF [38] because it uses Mip-NeRF as a baseline and possesses an outstanding ability to model glossy materials.

We also compare our method with NeRFReN on the RFFR dataset [12]. NeRFReN is a specially designed two-branch network based on vanilla NeRF for mirror-like surfaces in forward-facing scenes. Thus we construct a tiny version of our method, referred to as MS-NeRF\(_F\) based on NeRF with hyperparameters \( \{ K = 2, d = 128, h = 128 \} \). Here we use two sub-spaces as NeRFReN tries to decompose reflective surfaces into two parts, and we want to show that our space decomposition is more effective. For a fair comparison, we re-train NeRFReN using the official code under the provided settings on the RFFR dataset, except that we set the number of the used mask to 0 as our method requires no extra mask.

All the training details can be found in the supplementary. We report our results with three commonly used metrics: PSNR, SSIM [41], and LPIPS [51].

<table>
<thead>
<tr>
<th></th>
<th>PSNR↑</th>
<th>SSIM↑</th>
<th>LPIPS↓</th>
<th># Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeRF</td>
<td>30.82</td>
<td>0.865</td>
<td>0.209</td>
<td>1.159M</td>
</tr>
<tr>
<td>MS-NeRF(_S)</td>
<td>32.39</td>
<td>0.872</td>
<td>0.201</td>
<td>1.201M</td>
</tr>
<tr>
<td>MS-NeRF(_M)</td>
<td>32.61</td>
<td>0.875</td>
<td>0.195</td>
<td>1.245M</td>
</tr>
<tr>
<td>MS-NeRF(_B)</td>
<td>32.77</td>
<td>0.876</td>
<td>0.195</td>
<td>1.311M</td>
</tr>
<tr>
<td>Mip-NeRF</td>
<td>31.42</td>
<td>0.874</td>
<td>0.215</td>
<td>0.613M</td>
</tr>
<tr>
<td>Ref-NeRF</td>
<td>32.37</td>
<td>0.882</td>
<td>0.180</td>
<td>0.713M</td>
</tr>
<tr>
<td>MS-Mip-NeRF(_S)</td>
<td>33.63</td>
<td>0.886</td>
<td>0.195</td>
<td>0.634M</td>
</tr>
<tr>
<td>MS-Mip-NeRF(_M)</td>
<td>33.80</td>
<td>0.887</td>
<td>0.193</td>
<td>0.656M</td>
</tr>
<tr>
<td>MS-Mip-NeRF(_B)</td>
<td>33.90</td>
<td>0.888</td>
<td>0.191</td>
<td>0.689M</td>
</tr>
<tr>
<td>Mip-NeRF 360</td>
<td>31.58</td>
<td>0.895</td>
<td>0.145</td>
<td>9.007M</td>
</tr>
<tr>
<td>MS-Mip-NeRF 360</td>
<td>35.04</td>
<td>0.906</td>
<td>0.130</td>
<td>9.052M</td>
</tr>
</tbody>
</table>

Table 2. Quantitative comparisons with existing methods.
5.2. Comparisons

Quantitative comparisons. As reported in Tab. 2a, our module can be integrated into most NeRF-like models and improve performance by a large margin with minimal extra cost introduced. Especially in Mip-NeRF 360-based experiments, our module exhibits better results of 3.46 dB improvement in PSNR with merely 0.5% extra parameters. Besides, our Mip-NeRF-based models also outperform Ref-NeRF [38] by a large margin, which is a variant based on Mip-NeRF with the outstanding ability to model glossy materials. We also demonstrate our results compared with the state-of-the-art NeRFReN [38], which requires accurately labeled masks of the reflective regions during training and handles forward-facing reflective surfaces only. In this comparison, we train their model on our synthesized dataset with extra accurate reflection masks provided. Fig. 10 shows that their model fails to recover 360-degree high-fidelity rendering while our approach succeeds.

Qualitative comparisons and discussions. Besides quantitative comparisons, we summarize the advantages of our modules and support them by qualitatively or quantitatively comparing our methods with the corresponding baselines. Qualitative comparisons with the state-of-the-art method (i.e., Mip-NeRF 360) are shown in Fig. 1, Fig. 7, and Fig. 8. Our method renders high-fidelity virtual images, bounded and unbounded, in both synthetic and real-world scenes.

A qualitative comparison with Ref-NeRF [38], which understands virtual images as textures using the conventional NeRF backbone, is shown in Fig. 9. As Ref-NeRF is also based on Mip-NeRF, we compare our Mip-NeRF-based variant with Ref-NeRF using the same baseline and use comparable parameters (specifically ours 0.689M and Ref-NeRF 0.713M) in the comparison. Again the qualitative results show our significant improvements in rendering reflective surfaces.

We also compare with the NeRFReN model, which requires accurately labeled masks of the reflective regions during training and handles forward-facing reflective surfaces only. In this comparison, we train their model on our synthesized dataset with extra accurate reflection masks provided. Fig. 10 shows that their model fails to recover 360-degree high-fidelity rendering while our approach succeeds.

5.3. Ablation studies

In this section, we evaluate the design of our module and explore the relation between the number of sub-space and the number of virtual images.
Ablation on using neural feature field. We implement a module that simply outputs $K$ scalars $\{\sigma_k^i\}$ and $K$ RGB vectors $\{c_k^i\}$ of three dimensions, then we use the same integral equation as NeRF to get the pixel color of each sub-space and we average among all sub-spaces to get the final pixel color. We integrate this design into vanilla NeRF noted as MS-NeRF Avg, where we set $K = 6$, and the results are reported in Tab. 3. We also exhibit a few visual results in Fig. 11, which indicate that a simple multi-space radiance field assumption can help the model partially overcome the violation of reflections, but will also introduce the over-smoothing problem because of the lack of an efficient multi-space composition strategy.

Ablation on the sub-space number. In our Euclidean space, one can control the number of virtual sub-spaces by the number and the layout of the mirror(s). For example, when two mirrors are facing each other, there could be infinitely recursive virtual image spaces, but when two mirrors are placed back against each other, there will be just one virtual image behind each mirror. To provide a guideline for the design of our module, we choose two scenes consisting of two mirrors with different layouts from our synthesized part of the dataset and train NeRF-based variants of different sub-space numbers and different feature dimensions.

We construct our variants based on NeRF with the output feature dimensions $d \in \{24, 48, 64\}$ and the number of sub-spaces $K \in \{2, 4, 6, ..., 16\}$, then we train our models on the two scenes and report the results using PSNR.
References


[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG, 41(4), jul 2022. 2


